A Frequency-Based Learning-To-Rank Approach for Personal Digital Traces

Daniela Vianna
Rutgers University, New Jersey, USA
dvianna@gmail.com

Amélie Marian
Rutgers University, New Jersey, USA
amelie.marian@rutgers.edu

Abstract

Personal digital traces are constantly produced by connected devices, internet services and interactions. These digital traces are typically small, heterogeneous and stored in various locations in the cloud or on local devices, making it a challenge for users to interact with and search their own data. By adopting a multidimensional data model based on the six natural questions — what, when, where, who, why and how — to represent and unify heterogeneous personal digital traces, we propose a learning-to-rank approach using the state of the art LambdaMART algorithm and frequency-based features that leverage the correlation between content (what), users (who), time (when), location (where) and data source (how) to improve the accuracy of search results. Due to the lack of publicly available personal training data, a combination of known-item query generation techniques and an unsupervised ranking model (field-based BM25) is used to build our own training sets. Experiments performed over a publicly available email collection and a personal digital data trace collection from a real user show that the frequency-based learning approach improves search accuracy when compared with traditional search tools.

1. Introduction

Digital traces of our lives are constantly being produced and saved by users, as personal files, emails, social media interactions, multimedia objects, calendar items, contacts, GPS tracking of mobile devices, records of financial transactions, etc. Digital traces are usually small, heterogeneous and stored in various locations in the cloud or on local devices making it hard for users to access and search their own data.

Search of personal data is usually focused on retrieving information that users know exists in their own dataset, even though most of the time they do not know in which source or device they have seen the desired information. For instance, a user (John) might want to find the name and address of the restaurant he had lunch with his sister Anna when attending a conference in Paris some time in 2018, to recommend it to a friend. John remembers going to the restaurant, but does not know where relevant information may be recorded in his personal data: an email, a calendar entry, a tweet, a credit card transaction, or in a photo. Personal search have been previously studied in specific real-life scenarios such as desktop search [1] and email search [2]. In this work, we focus on search over an integrated personal dataset comprised of a multitude of heterogeneous personal digital traces from a variety of data sources. In this setting, each personal digital trace is a source of knowledge and can be related to different data traces by shared common information, modeled following the six contextual dimensions: *what*, *who*, *when*, *where*, *why* and *how*. The richness of contextual information attached to the digital data proves to be of great help to users searching for information they remember having stored and accessed in the past.

Searching personal data requires ranking the best potential answers based on their relevance. Learning-to-rank approaches have been very successful in solving real-world ranking problems. However, existing models for ranking are trained on either explicit relevance judgments (crowdsourced or expert-labeled) or clickthrough logs, which are typically not available for personal data. In addition, there is a dearth of synthetic personal data sets and benchmarks. To overcome those challenges, we propose a learning-to-rank approach that relies on a combination of known-item query generation techniques and an unsupervised ranking model (field-based BM25) to heuristically build training sets. Furthermore, since personal digital traces are results of actions and events of users, the correlation, i.e., the relationships or any type of connections within and across traces between objects, can be leveraged to improve the accuracy of search results. We represent the input data by a set of frequency-based features that takes into

URI: https://hdl.handle.net/10125/79695
978-0-9981331-5-7

consideration the correlation between content (*what*), users (*who*), time (*when*), location (*where*) and data source (*how*). We use a state-of-the-art learning-to-rank algorithm based on gradient boosted decision trees, LambdaMART [3] to learn a ranking model to map feature vectors to scores. The work presented in this paper is developed as part of a series of tools to let user retrieve, store and organize their digital traces *on their own devices* guaranteeing some clear privacy and security benefits.

In this paper, we present the following contributions: (1) a feature set to represent query-matching object pairs over personal digital traces; our features are built upon a novel frequency-based feature space that leverages entities interactions within and across dimensions in the dataset, (2) a novel combination of known-item query generation techniques and an unsupervised ranking model to heuristically generate labeled training sets for personal digital traces, (3) a quantitative evaluation of the proposed search techniques, as well as comparison with two popular search methodologies: *BM25* and *field-based BM25* (*BM25-f*). Our results show that moderately large personal datasets can benefit from learning techniques when combined with a compact frequency-based feature set.

2. Data Model

Based on the observation that contextual cues are strong trigger for autobiographical memories ([4, 5, 6, 7]), and that personal data is rich in contextual information, in the form of metadata, application data, or environment knowledge, we can represent personal digital traces using a combination of dimensions that naturally summarize various aspects of the data collection: *who*, *when*, *where*, *what*, *why* and *how*. Our work uses an intuitive multidimensional data model that relies on these six dimensions as the unifying features of each personal digital trace object, regardless of its source [8]:

what: content such as messages, messages subjects, description of events, list of interests of a user

who: user names, senders, recipients, event owners

where: physical or logical, in the real-world and in the system. For instance, hometown, location, event venues, URLs, file/folder paths

when: time and date, but also what was happening concurrently. Example, birthday, file/message/event created-/modified- time

why: sequences of data/events that are connected

how: application, author, environment

Figure 1a shows two personal digital traces, an email message and a calendar entry, relating to our user John's

```
From: John Smith                                  (WHO)
To: Anna Smith                                    (WHO)
Date: 2018-09-04T10:30:00+0000                    (WHEN)
Subject: Lunch                                    (WHAT)
Body: Do you want to get something to eat?        (WHAT)
```

(a) User email message

```
Description: Conference Center.                   (WHAT)
Creator.displayName: Anna Smith                   (WHO)
Location: Paris, France                           (WHERE)
Attendees.displayName: John Smith                 (WHO)
Created: 2018-09-05T11:00:00+0000                 (WHEN)
Data_type: Calendar Event                         (HOW)
```

(b) Google Calendar entry

Figure 1: Simplified examples of a user email message and a calendar entry classified according to the 6 contextual dimensions model.

trip to Paris. Each piece of information in the traces can be classified as belonging to one of the six contextual dimensions: *what, who, where, when, why and how*. Using these six dimensions, we can unify and link multiple digital traces that might come from different sources and have their own data schema.

Having defined the multidimensional data model, it is still necessary to find an effective mechanism to automatically translate the heterogeneous set of personal data into the six dimensions. A simple approach would be to use ETL rules, but this proved unpractical as new sources of data were added, and the schemas of the existing sources were modified by the third-party apps. Therefore, we opted for a machine learning multi-class classifier using a combination of LSTM (Long Short-Term Memory) and Dense layers [8]. Given a sentence, the classifier will output a label (*who*, *when*, *where*, *what*, *why* and *how*). For instance, the subject in an email is a sentence with label *what*. During the classification process, the *when* dimension is normalized allowing dates from different sources to be matched. Normalization also makes possible partial matches. As an example, a query searching for June, will match objects with time June 2016 and June 2017 regardless of the original format of the data. Entity resolution is applied to the parsed *who* and *where* dimensions identifying separate instances of the same entity in data traces coming from the same sources, and across sources. We use the classifier to translate raw data retrieved from third-party sources into the 6 dimension model, without the need of human intervention.

3. Scoring Model

Unlike Web search, where the focus is often on discovering new relevant information, search in personal datasets is typically focused on retrieving data that the user knows exists in their dataset. Furthermore, users have unique habits and interpretations of their own data.

In such a setting, standard search techniques are not ideal as they do not take advantage of the additional knowledge the user is likely to have about the target object, or the connections between objects pertaining to a given user. This extra knowledge, represented by the six dimensional model introduced in Section 2, can be leveraged to provide accurate search capabilities over personal digital traces.

Personal digital traces are very specific to each user and are constantly evolving over time, it is thus necessary to find a scoring model that can generalize well over user-specific datasets. Learning-to-rank approaches have proved to be very efficient to solve ranking problems, however, learning algorithms require a large amount of data to generalize well given the large feature space usually employed (from thousands to tens of thousands of features), due to the curse of dimensionality [9]. To be able to employ a learning approach on available personal datasets, which are typically not very large, we adopted a compact feature set based on a representation of the input data using the six contextual dimensions presented in Section 2.

In Section 3.1, we define matching objects and queries in our personal search scenario. The 34 features proposed are detailed in Section 3.2. In Section 3.3 we continue to discuss the frequency score focusing on the *what* dimension that is comprised mostly of text. Section 3.4 describes the learning-to-rank algorithm used in this work to generate our ranking model and validate the feature space proposed.

3.1. Scoring Methodology

We consider each digital trace from a personal dataset to be a distinct object that can be returned as the result to a query.

Definition 1 (Object in the Integrated Dataset) *An object O in the dataset is a structure that has fields corresponding to the 6 dimensions mentioned earlier. Each of these dimensions contains 0 or more items (corresponding to text, entities identified by entity resolution, times, locations, etc).*

Definition 2 (Query) *A query Q over the dataset is represented as an object as defined in Definition 1.*

Given objects Q and O, O is considered as an answer to object Q treated as a query if it contains at least one dimension/item specified in Q. Unfortunately users' memories are notoriously unreliable [6, 10], and fully trusting their recollection of contextual information can lead to miss relevant results. In looking for (partially) matching objects to a given query, each dimension will be searched separately, and the results will be combined generating a list of candidates with some partial order. In order to find an optimal order for this list of candidates, we introduce our learning-to-rank approach on top of a representative feature set built from our novel frequency-based feature space.

3.2. Frequency-based Features

Because personal digital traces are byproducts of actions and events of users, they are not independent objects. Our intuition is that the correlation between traces (objects) can be leveraged to improve the accuracy of search results. We explore how the correlations between users (*who*), time (*when*), location (*where*), topics (*what*) and data sources (*how*) can be used to improve search over personal data. We exploit those interactions and correlations by way of a frequency score. For each dimension and combination of dimensions we compute a score that will be used later as features to represent the input data in our learning-to-rank approach.

Frequencies can be computed at different granularities: individual users or group of users, multiple time intervals, multiple data sources, locations. Frequency expresses the strength of relationships, based on users, time, location, content and data sources (*who, when, where, what, how*). For each object in the data set, the frequency algorithm considers the information associated with each dimension to compute the frequencies. For example, the frequency of each individual user in a data set is the number of objects that mention that user in the *who* dimension. Similarly, the frequency of each individual user at specific times is the number of objects that mention that user (*who*) at matching times (*when*).

To take advantage of the strong correlation between group of users, which is an important feature of personal corpora, we also compute the frequency between group of users, source, times and location. For instance, the frequency of a group of users at specific times is the number of objects mentioning the group (*who*) at a specific time (*when*). We use a set of 34 features to represent the input data. The feature set is comprised of 30 features resulting from all possible combinations between the dimensions *who, what, when, where* and *how* plus 4 extra features that model the correlation between group of users (*who groups*); group of users and time (*who groups, when*); group of users and data source (*who groups, how*); and finally, group of users, time and location (*who groups, when, where*). The feature vector is defined in Definition 3.

Definition 3 (Feature Vector) $\boldsymbol{x} = [x_1 \dots x_{34}]$ *is a feature vector comprised by 34 frequency-based*

features. Each feature x_i is computed by a frequency function $f(S_i, Q, O)$, where $S_i \in \mathcal{S}$. $\mathcal{S}$ represents all possible combinations between the 5 dimensions who, what, when, where *and* how*, plus the 4 extra features that model the correlation between group of users. Q is a query (Definition 2) and O is an object in the user dataset (Definition 1).*

To illustrate our query and scoring methodology consider the following search scenario: the user is interested in a message from 2018 (*when*), sent by John (*who*), about the topic "Lunch" (*what*). We can define query Q_1 as (*when*: 2018, *who*: John, *what*: Lunch). By Definition 1, the object in Figure 1a (O_1) is a matching to the given query (Q_1) containing all dimension/item specified in the query – *when*:2018, *who*:John, and *what*:"Lunch". The query-object pair (Q_1, O_1) can be represented by a 34 frequency-based feature vector $\mathbf{x} = [x_1 \dots x_{34}]$ as introduced in Definition 3. Each feature x_i represents the frequency score for a set of dimensions S_i, query Q_i and object O_i:

$x_1 = f((\text{what:Lunch}), Q_1, O_1)$
$x_2 = f((\text{who:John}), Q_1, O_1)$
$x_3 = f((\text{when:2018}), Q_1, O_1)$
$x_6 = f((\text{what:Lunch, who:John}), Q_1, O_1)$
$x_7 = f((\text{what:Lunch, when:2018}), Q_1, O_1)$
$x_9 = f((\text{what:Lunch, how:Gmail}), Q_1, O_1)$
$x_{10} = f((\text{who:John, when:2018}), Q_1, O_1)$
$x_{12} = f((\text{who:John, how:Gmail}), Q_1, O_1)$
$x_{16} = f((\text{what:Lunch,who:John, when:2018}), Q_1, O_1)$
$x_{18} = f((\text{what:Lunch,who:John, how:Gmail}), Q_1, O_1)$
$x_{20} = f((\text{what:Lunch,when:2018, how:Gmail}), Q_1, O_1)$
$x_{23} = f((\text{who:John, when:2018, how:Gmail}), Q_1, O_1)$
$x_{27} = f((\text{what:Lunch,who:John,when:2018,how:Gmail}), Q_1, O_1)$

If a set of dimensions S_i is not present in query Q_i and object O_i, the frequency score $f(S_i, Q_i, O_i) = 0$. When a query Q contain multiple values for a dimension e.g., Q = (who: John, Alice), the values for each feature x_i will be the summation of the frequencies for each individual value.

To understand how frequencies ($f(S_i, Q, O)$) are computed, consider the following example: lets assume a dataset D containing 10 objects that mention John under the *who* dimension, being 4 of those 10 objects from Facebook and the remaining 6 from Gmail. Given object O_1 and query Q_1 from the previous example, we can say that the frequency of John ((who:John)) in dataset D for query Q_1 and matching object O_1 is $x_2 = f((\text{who:John}), Q_1, O_1)$, where $f((\text{who:John}), Q_1, O_1) = 10$. We can also say that the frequency of John in Gmail ((who:John,how:Gmail)) in dataset D for query Q_1 and matching object O_1 is $x_{12} = f((\text{who:John, how:Gmail}), Q_1, O_1)$, where $f((\text{who:John, how:Gmail}), Q_1, O_1) = 6$.

Notice that the *why* dimension is not explored in this paper and only included for completeness, The *why* dimension is the topic of related work [11, 12] that use inference to connect different fragments of data that derive from a common real-life task, or episode (e.g., all traces that stemmed from a restaurant outing).

3.3. Scoring the *What* Dimension

The *what* dimension in the six-dimension model is composed of content information comprising mostly of text. We use two standard text approaches to link and score objects for the *what* dimension: field-based BM25 (used to score the *what* dimension alone) and topic modeling [13] (used to link the *what* dimension with the other dimensions).

Field-based BM25. A field-based BM25 is a state-of-the-art TF-IDF type of ranking function that takes into consideration the document structure. In our scenario, the fields in the field-based BM25 correspond to the 5 dimensions proposed, *what, who, when, where* and *how*. To compute the field-based BM25 score for the *what* dimension, we use a popular full-text search platform from the Apache Lucene project, Solr[1], with its default parameters. All data retrieved for a user is unified and parsed according with the six dimensions (Section 2) and then, exported to Solr. For each user query, we search Solr using the values from the *what* dimension, getting as a result a partial list of matching documents with its respective field-based BM25 score. Even though Solr contains the data for all 5 dimensions, we are only interested in use field-based BM25 to score the *what* dimension, since this dimension contains most of the content of an object. For the remaining dimensions, we use our frequency-based function as introduced in Section 3.2.

Topic Modeling. A "Topic" consists of a cluster of words that frequently occur together. Topic models use contextual cues to find connections between words with similar meanings and to distinguish between use of words with multiple meanings. Given a document, we would like to identify what possible topics have generated that data. In our case, topic modeling would be an important feature to connect different objects, including objects from different data sources. The association between topics (*what*), user (*who*), times (*when*), location (*where*) and source (*where*) could shed some light on finding objects that could be a better matching to the user query. To define

[1] http://lucene.apache.org/solr/

topics for each object in the user data set, we use a topic model package called MALLET and a text collection built from the content classified under the *what* dimension for each object in the user data set. The MALLET [14] topic model package includes a fast and scalable implementation of Latent Dirichlet Allocation (LDA) with Gibbs sampling. For each object in the user data set, MALLET computes the topic composition of documents. We use the default MALLET hyper parameters, except for number of topics parameter, that we set to 50 based on the size of the data set and a visual inspection of the topics generated for different number of topics varying from 10 to 100. We use the most relevant topic for each document, i.e., the one with the highest composition percentage, to cluster documents per topic. For each document in a topic, we extract the person/entity mentioned in *who* dimension, the times from *when*, location from *where* and source from *how*. Using this information, we are able to build the correlation between person/entity, times, location and source for each topic (*what*), and to estimate the frequency of those correlation/interactions using the frequency function presented in Section 3.2.

MALLET also provides a list of the words in the documents of corpus with their topic assignments and frequencies. We use this information in conjunction with the words specified in the user query (for the *what* dimension), to find the topic that are closest to the user query. This allows us to narrow down a partial list of documents that are matching candidates to the query, based solely on the contents of the *what* dimension.

To illustrate how topic modeling can support our search, consider T a topic composed by the following key words: hotel, lunch, street, trip, miles, view, lake, ride, restaurant and conference. Assuming that topic T is the most relevant topic for object O_1 in Figure 1a, and object O_2 in Figure 1b, we can say that objects O_1 and O_2 are correlated by their *what* dimension. By considering all objects (documents) clustered under the same topic T, we can learn how strong person/entity (*who*), times (*when*), location (*where*) and source (*how*) are connected with relation to a topic (*what*). This strength is measured by a way of a frequency score as presented in Section 3.2.

3.4. Learning-to-Rank Model

In the previous sections, we explained how query-document pairs are represented by a feature vector built upon our frequency-based feature space. To map the feature vector to a real-valued score we need to train a ranking model. Our choice of learning-to-ranking algorithm is the state-of-the-art LambdaMART [3]. LambdaMART uses gradient boosted decision trees, which incrementally builds regression trees trying to correct the leftover error from the previous trees. At the end, the prediction model is an ensemble of weaker prediction models that complement each other for robustness. During a training phase, we must define the best set of parameters that results in a robust and accurate model. For this cross-validation stage, we will consider the following parameters: number of trees in the ensemble, maximum number of leaves per tree, minimum number of samples each leaf has to contain, learning rate (shrinkage), and training metric.

4. Query Sets

In a learning-to-rank algorithm, each pair of query-document(object) is represented by a vector of numerical features. In addition to the feature vector, pairs of query-documents could be augmented with some relevance information. Then, a model has to be trained to map the feature vector to a score. One of the challenges of using learning-to-rank for personal data search is to be able to build a training set without human intervention or any external information (e.g., expert labeling or click data). To this end, in this section we present a combination of heuristics that given a user dataset is able to simulate a human-labeled training set to tailor the learning model to each specific user dataset.

Search of personal data is usually focused on retrieving information that users know exists in their own data set. Considering the fact that personal data search is a known-item type of search, simulated queries can be automatically generated, using known-item query [15] generation techniques such as the ones presented in [16] and [17]. In this work, queries are created by randomly choosing a set of dimensions (*who, what, when, where, how*) and values/items (e.g. email's Subject, Facebook post's content) from a target object, as described in Algorithm 1. Each call to Algorithm 1 will result in a query-target object pair.

By using the proposed known-item query generation technique, we are able to build a list of query-target object pairs. However, a learning-to-rank training set is composed not only by pairs of query-known document, but also by a list of matching documents per query. In [18], the authors use classic unsupervised information retrieval models, such as BM25, as a weak supervision signal for training deep neural ranking models. In a similar fashion, we adopt an unsupervised ranking model, field-based BM25, to retrieve matching objects to a given query. In Section 3.3, we explained how the data retrieved for a user is unified and parsed according with the six dimensions (Section 2) and then, the parsed

Algorithm 1 Known-item query generation algorithm.

```
1: procedure BUILD–QUERY(DATASET D)
2:     Q = () /* Initialize query Q. */
3:     /* Randomly choose a target object O_i from the
       dataset D */
4:     O_i = random(D)
5:     /* Select dimensions */
6:     d = select_dimensions({what, who, when,
       where, how})
7:     for each d_i ∈ d do
8:         /* Randomly choose v values from target object
           O_i and dimension d_i */
9:         v = select_values(d_i)
10:        /* Add dimension and values to the query Q */
11:        Q(d_i) = v
12:    end for
13:    return Q
14: end procedure
```

data is exported to Solr where it can be searched using a field-based BM25 approach. Given a query generated by Algorithm 1, a call to Solr will retrieve a list of matching documents to this query — the list is ranked using field-based BM25. Now, for each query, we have a list of matching documents that includes the (generated) target object and its corresponding feature vector as described in Section 3.2. Since the target object is known for this query, a relevance label of 1 is assigned to it; otherwise, the relevance label will be 0.

5. Case Studies 1: Personal Digital Data Traces

We evaluate the effectiveness of the proposed ranking model by comparing its results with two popular scoring methodologies: *BM25* and *field-based BM25* (BM25-f). Experiments are performed over real user data from a variety of data sources.

Data Set. There is a dearth of synthetic data sets and benchmarks to evaluate search over personal data. Thus, we perform our evaluation using a real dataset collected by a personal data extraction tool [19] containing approximately two hundred thousand objects. Table 1 shows the composition of our real user dataset, including the number and size of objects retrieved from different sources over different periods of time. The dataset was automatically classified according with the 6 contextual dimensions (Section 2).

The data set was collected in 2014, since then third-party APIs have changed significantly and some of the data available through APIs then is not accessible anymore (e.g., Facebook). It is however important to consider that a plethora of new service, social and otherwise, have been created since, and users' personal digital traces are created and collected at an increasing rate. Many of these services allow users to download and store their own data, often to fulfill legal and regulatory requirements. Our model and techniques can be applied to a large array of data types.

Data Source	#Objs	Size
Facebook	3875	28Mb
Gmail	28318	3Gb
Dropbox	573	32Mb
Foursquare	55	59Kb
Twitter	3929	22Mb
Google Calendar	330	620Kb
Google+	110	367Kb
Google Contacts	525	629Kb
Bank	412	415Kb
Firefox	181921	63Mb
Total	219,993	3.6Gb

Table 1: Personal dataset.

Training and Evaluation query sets. We train and evaluate our model on the user's own device using heuristically generated samples. As detailed in Section 4, each query is automatically created by randomly choosing a target object from the data set. We then choose d dimensions, from which we randomly select v random values. For this set of experiments, we built a training set comprised by 19000 queries over our personal dataset (Table 1). To built the query sets, we use $v = 1$ and 4 different values for parameter d: {*what, who*}, {*what, who, when*}, {*what, who, when, how*}, and {*what, who, how*}. The evaluation set was built in a similar fashion. Approximately 6000 queries were heuristically generated using the same combination of parameters as the training set. Since less than 2% of objects in the user dataset have location, the dimension *where* was not included in the query sets. Examples of queries used in the evaluation are: Q1: *what*: Databases, *who*: Entity Alice, *when*: 2017-06; Q2: *what*: Meditation, *who*: Entity Jerry, *when*: 2016, *how*: Gmail.

Evaluation Techniques and Metrics. We evaluate the efficacy of the proposed approach by comparing it with two popular scoring methodologies: *BM25* and *field-based BM25*.

BM25 is a state-of-the-art type of TF-IDF function that ranks a list of matching documents based on the query content that appears in each document. To be able to use BM25 with the retrieved dataset (Section 5), the decentralized digital traces have to be integrated in one unified collection. It is done by exporting the data retrieved to a unified data collection in Solr. This approach allows user to search for information across

the entire set of retrieved digital traces, which is already a significant step forward from the current state, where users have to search each data source individually.

Field-based BM25 is a version of BM25 that takes into consideration the structure of a document. In our scenario, the fields in the field-based BM25 correspond to the five dimensions proposed: *what, who, when, where, how*. As described in Section 3.3, before being exported to Solr, the retrieved dataset (Section 5) is unified and parsed according with the six dimensions (Section 2). It allows for the dataset to be searched using field-based BM25 with each field corresponding to a respective dimension. Note that by using the five dimensions, **we are giving the field-based BM25 approach the advantage of using our multidimensional data model to unify and organize the user data**.

The scoring model proposed is evaluated using 4 standard evaluation metrics: Mean Reciprocal Rank (MRR) of the top-ranked 50 documents, precision of the top 1 retrieved document (p@1), precision of the top 3 retrieved document (p@3), and precision of the top 10 retrieved document (p@10). Wilcoxon signed-rank test with $p\_value < 0.05$ is used to determine statistically significant differences. Statistically significant results are market with * in the tables.

Ranking Model. To train and evaluate our model, we use the LambdaMART implementation provided by the RankLib library[2]. RankLib is a library of learning to rank algorithms that is part of The Lemur Project[3]

The first step in our evaluation was to define the best set of parameters that would give us a more robust and accurate model. The parameters evaluated are: number of trees (50, 100, 250 and 500), number of leaves for each tree (10, 15, 35 and 45), minimum leaf support (10, 20 and 50), shrinkage (0.01, 0.03, 0.1, 0.3, 0.5, 1.0), and metric (Mean Reciprocal Rank).

We adopted a 5-fold cross validation process to estimate the performance of each model. After the validation process, we selected the model that shows the best performance on the training set. The model selected, that we will call *w5h-l2r*, has the following parameters: number of trees $= 50$; number of leaves $= 15$; minimum leaf support $= 10$; shrinkage $= 0.1$.

In Table 2 we compare the ranking performance of the baseline (*BM25*), *field-based BM25* (*BM25-f*) and learned ranking model (*w5h-l2r*) with respect to the entire evaluation set composed by approximately 6000 queries heuristically generated as described in Section 5. The results show that both search models using the data parsed according to the multidimensional data model (Section 2), *field-based BM25* and *w5h-l2r*, outperform the keyword-based approach, *BM25*, for MRR, p@1, p@3 and p@10. It shows that traditional keyword-based search methods are not appropriate in a setting where users may remember valuable contextual cues to guide the search. Observe that the *w5h-l2r* model, outperform the *field-based BM25* approach for all 4 evaluation metrics, **showing that moderately large datasets can also benefit from learning-to-rank techniques when paired with a representative feature set built from our novel frequency-based feature space**.

Method	MRR	p@1	p@3	p@10
BM25	0.363	0.270	0.410	0.535
BM25-f	0.508*	0.425*	0.550*	0.669*
w5h-l2r	0.518*	0.441*	0.560*	0.690*

Table 2: MRR, p@1, p@3, p@10 for all 6000 queries (groups 1 to 4) from the Personal Digital Data dataset.

We now conduct a more thorough evaluation by dividing the evaluation set (Section 5) in four different groups by the dimensions in each query: Group 1 = what, who; Group 2 = what, who, when; Group 3 = what, who, when, how; Group 4 = what, who, how.

Table 3a-d, show the MRR, p@1, p@3 and p@10 of each search approach, *BM25* (baseline), *field-based BM25*, and *w5h-l2r*, for Group 1 to 4 of queries. For all 4 groups, the search approaches that use the data classified according with our multidimensional data model are considerably more accurate than the keyword-based approach, *BM25*, confirming the importance of including contextual information to improve search accuracy when searching personal data. When compared against each other, *field-based BM25* and *w5h-l2r*, the learned ranking model outperform the *field-based BM25* model, *BM25-f*, for all four groups; however, the improvements were more relevant for Group 2 (*what, who, when*) and Group 3 (*what, who, when, how*), showing that for this dataset, using the proposed learning model and training data, the *when* dimension and all related features played an important role in scoring query-document pairs. The results for *w5h-l2r* when compared with *BM25-f* are statistically significant (Wilcoxon signed-rank test, $p\_value < 0.05$) for Groups 2 and 3, evaluation metric MRR and p@k. For Group 1 the results are not statistically significant for MRR and p@3. For Group 4, the results are not statistically significant for MRR, p@1 and p@3.

Figure 2 presents the performance (MRR) of the *w5h-l2r* model, for Group 1 of queries as the number of training samples increases. The effectiveness of the learned ranking model (*w5h-l2r*) clearly improves as the size of the training set increases just modestly. The same

[2] http://www.lemurproject.org/ranklib.php
[3] http://www.lemurproject.org

Method	MRR	p@1	p@3	p@10
BM25	0.344	0.245	0.394	0.529
BM25-f	0.441⋆	0.361⋆	0.481⋆	0.600⋆
w5h-l2r	0.443⋆	0.373⋆	0.485⋆	0.643⋆

(a) Group 1: what, who

Method	MRR	p@1	p@3	p@10
BM25	0.376	0.288	0.422	0.537
BM25-f	0.576⋆	0.485⋆	0.626⋆	0.742⋆
w5h-l2r	0.597⋆	0.508⋆	0.647⋆	0.766⋆

(b) Group 2: what, who, when

Method	MRR	p@1	p@3	p@10
BM25	0.421	0.322	0.461	0.587
BM25-f	0.617⋆	0.522⋆	0.642⋆	0.781⋆
w5h-l2r	0.633⋆	0.527⋆	0.662⋆	0.794⋆

(c) Group 3: what, who, when, how

Method	MRR	p@1	p@3	p@10
BM25	0.348	0.259	0.389	0.520
BM25-f	0.462⋆	0.386⋆	0.497⋆	0.615⋆
w5h-l2r	0.463⋆	0.396⋆	0.490⋆	0.600⋆

(d) Group 4: what, who, how

Table 3: MRR, p@1, p@3, p@10 for groups 1, 2, 3, and 4 from the Personal Digital Data Traces dataset (Table 1).

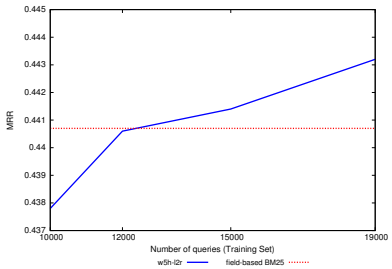

Figure 2: Performance (MRR) of the learning model, *w5h-l2r*, for group 1 of queries from the Table 1 dataset as the number of training samples increases.

trend was observed for Groups 2, 3, and 4 of queries.

The importance of a feature in a gradient boosted decision tree model such as LambdaMART can be conveyed by the number of times such feature appears in the internal (non-leaf) nodes of the decision trees. Since our model has 50 trees, each having 15 leaves (and 14 internal nodes), there are 700 branches overall. For the trained *w5h-l2r* model, the most frequent feature is the *what* dimension, that represents the content of an object. Then, features (*who,when*), (*who,how*), and (*who*) appear next, all of them related to **the *who* dimension, which is expected since personal traces are byproduct of actions and events of users (*who*), and are typically focused on user interactions**.

The results presented in this section indicates that personal data search can improve greatly by taking into consideration the knowledge the user has about the object being searched. The multidimensional data model proved to be an effective model to unify and link digital traces. The advantage of using a learning approach to re-rank search results can be seen by the improvement presented by the *w5h-l2r* approach when compared against both methods, *BM25* and *BM25-f*. Including a compact feature space based on frequency information resulted in significant improvements.

6. Case Studies 2: Enron Email Dataset

Data Set. To verify the validity of our learning-to-rank approach over other domains, we have implemented it over an email dataset: the Enron dataset[4]. The Enron email dataset contains a total of about 0.5M emails from 158 employees of the Enron Corporation, obtained by the Federal Energy Regulatory Commission after the company collapsed into bankruptcy.

Training and Evaluation Query Sets. We use heuristically generated samples as we did in Section 4. Each query is automatically created by randomly choosing a target object from the data set. We then choose d dimensions, from which we randomly select v random values. For this set of experiments, the training set is comprised by 48000 queries over the Enron dataset and the evaluation set is comprised by 2000 queries. Training and evaluation query sets were built using $v = 1$ and 4 different values for parameter d: {*what, who*}, {*what, who, when*}, {*what, who, when, how*}, and {*what, who, how*}.

Ranking Model. To train and evaluate our model, we use the same parameters and steps presented in Section 5. The model selected has the following parameters: number of trees = 50; number of leaves = 15; minimum leaf support = 20; shrinkage = 0.3.

As with the Personal Digital Data Traces dataset, for the Enron dataset the evaluation set was divided in four different groups by the dimensions in each query: Group 1 = what, who; Group 2 = what, who, when; Group 3 = what, who, when, how; Group 4 = what, who, how.

Table 4a-d, show the MRR, p@1, p@3 and p@10 of each search approach, *BM25* (baseline), *BM25-f*, and *w5h-l2r*, for Group 1 to 4 of queries. For Group 1 (Table 4a), the search approach *w5h-l2r* is slight better than *BM25* (baseline) and *BM25-f* for MRR and p@1. For Group 2 (Table 4b) and Group 3 (Table 4c), the search approaches that use the data classified according

[4] http://www.cs.cmu.edu/~enron/

Method	MRR	p@1	p@3	p@10
BM25	0.259	0.132	0.112	0.050
BM25-f	0.255	0.126	0.105	0.051
w5h-l2r	0.269	0.156	0.102	0.050

(a) Group 1: what, who

Method	MRR	p@1	p@3	p@10
BM25	0.235	0.122	0.098	0.045
BM25-f	0.414*	0.236*	0.177*	0.073*
w5h-l2r	0.422*	0.250*	0.179*	0.073*

(b) Group 2: what, who, when

Method	MRR	p@1	p@3	p@10
BM25	0.242	0.133	0.098	0.045
BM25-f	0.409*	0.231*	0.179*	0.074*
w5h-l2r	0.421*	0.258*	0.176*	0.074*

(c) Group 3: what, who, when, how

Method	MRR	p@1	p@3	p@10
BM25	0.244	0.106	0.109	0.048
BM25-f	0.259	0.114	0.113	0.049
w5h-l2r	0.248	0.122	0.102	0.049

(d) Group 4: what, who, how

Table 4: MRR, p@1, p@3, p@10 for groups 1,2,3, and 4 from the Enron dataset.

with our multidimensional data model are considerably more accurate than the keyword-based approach, *BM25*, and the results are statistically significant (Wilcoxon signed-rank test, $p\_value < 0.05$). For those groups, the learned ranking model, *w5h-l2r*, outperforms the *field-based BM25* model for all metrics, the exceptions being success@10 for Group 2 and success@3 for Group 3. For Group 4, all approaches had a similar performance. For most scenarios in this group of queries, the features based on the *how* dimension are not contributing to differentiate Enron results.

Looking at the feature frequency distributions for the learned ranking model *w5h-l2r*, we observe that the two most frequent features are based on the *what* dimension, (*what*) and (*what, how*), that represents the content of an object. Then, features related to the *who* dimension appear next, representing the frequency of users and the interactions between user/time and user/topic.

The results discussed in this section show that even though the data and scoring model were proposed with the Personal Digital Data Traces dataset in mind, it can be extended to different domains with promising results.

7. Related Work

Bell has pioneered the field of life-logging with the project MyLifeBits [20, 21] for which he has digitally captured all aspects of his life. *digi.me*[5] is a commercial tool that aims at extending Bell's vision to everyday users. The motivations behind *digi.me* are very close to ours; however *digi.me* currently only offers a keyword- or navigation-based access to the data; search results can be filtered by service, data type or/and date.

The case for a unified data model for personal information was made in [22, 23]. deskWeb [24] looks at the social network graph to expand the searched data set to include information available in the social network. Stuff I've Seen [1] indexes all of the information the user has seen, regardless of its location, and uses the corresponding metadata to improve search results. Our work is related to the wider field of Personal Information Management [5], in particular, search behavior over personal digital traces is likely to mimic that of searching data over personal devices. Unlike traditional information seeking, which focuses on discovering new information, the goal of search in Personal Information systems is to find information that has been created, received, or seen by the user.

Email search is a type of personal search that has been extensively studied. [2] presents a learning-to-rank approach that improves the default ranked-by-time search by taking into consideration time recency and textual similarity to the query. [25] addresses the problem of learning-to-rank from click data in personal search. [26] explores how to effectively leverage situational contextual features (e.g. time and location of a search request) to improve personal search quality. In [27] the authors leverage user interaction data in a privacy preserving manner for personal search by aggregating non-private query and document attributes across a large number of user interactions. In our scenario, each dataset contains data from only one user, and their interactions with others that they are already permitted to access, at no point does the search consider private data from other users.

In [18] the authors use classic unsupervised IR models as a weak supervision signal for training deep neural ranking models. In this context, weak supervision refers to a learning approach that creates it own training data by heuristically retrieving documents for a large query set. Three different neural network-based ranking models are presented, a point-wise and two pair-wise ranking models. Combinations of neural models with different training objectives and input representations are compared against each other and against the baseline, BM25. The experiments showed that their best performing model significantly outperforms the BM25 model. In our work, we use a similar approach to retrieve matching objects to a given query.

[5] https://www.digi.me

8. Conclusions and Future Work

In this work, we proposed a learning-to-rank approach that uses a compact and efficient frequency-based feature space to rank query results over personal digital traces. The learning model relies on our multidimensional data model to unify and link heterogeneous digital traces using six contextual dimensions: *what, who, when, where, why* and *how*. To overcome the lack of human-labeled training sets, we proposed a combination of known-item query generation techniques and an unsupervised ranking model (*field-based BM25*) to generate our query sets. Experiments over a publicly available email collection and a personal dataset composed by data from a variety of data sources indicates that our frequency-based learning approach can significantly improve the accuracy of search results when compared with a traditional keyword-based approach, *BM25*, and a field-based approach that uses the data parsed according to our multidimensional data model, *field-based BM25*.

References

[1] S. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins, "Stuff ive seen: A system for personal information retrieval and re-use," in *Proceedings of the 26th International ACM SIGIR Conference (SIGIR'03)*, 2003.

[2] G. Halawi and A. Raviv, "Rank by time or by relevance?: Revisiting email search," in *Proceedings of the 24th ACM Conference on Information and Knowledge Management*, 2015.

[3] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," tech. rep., June 2010.

[4] W. Brewer, *Memory for randomly sampled autobiographical events*, p. 21–90. Cambridge University Press, 1988.

[5] W. Jones and J. Teevan, eds., *Personal Information Management*. University of Washington Press, 2007.

[6] D. Schacter, *The seven sins of memory: How the mind forgets and remembers.* Houghton Mifflin, 2001.

[7] W. A. Wagenaar, "My memory: A study of autobiographical memory over six years," *Cognitive Psychology*, vol. 18, no. 2, pp. 225 – 252, 1986.

[8] D. Vianna, V. Kalokyri, A. Borgida, A. Marian, and T. Nguyen, "Searching heterogeneous personal digital traces," in *ASIST'19: Proceedings of the 82nd ASIS&T Annual Meeting, Melbourne, AU*, 2019.

[9] R. E. Bellman, *Adaptive Control Processes*. Princeton University Press, 1961.

[10] W. B. Croft, P. Krovetz, and H. Turtle, "Interactive retrieval of complex documents," *Information Processing and Management*, vol. 26, no. 5, 1990.

[11] V. Kalokyri, A. Borgida, A. Marian, and D. Vianna, "Semantic modeling and inference with episodic organization for managing personal digital traces," in *Proc. 16th International Conference on Ontologies, DataBases, and Applications of Semantics.*, 2017.

[12] V. Kalokyri, A. Borgida, and A. Marian, "Yourdigitalself: A personal digital trace integration tool," in *Proc. 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 1963–1966, 2018.

[13] M. Steyvers and T. Griffiths, *Latent Semantic Analysis: A Road to Meaning*, ch. Probabilistic topic models. Laurence Erlbaum, 2007.

[14] A. K. McCallum, "Mallet: A machine learning for language toolkit." http://mallet.cs.umass.edu, 2002.

[15] D. Elsweiler and I. Ruthven, "Towards task-based personal information management evaluations," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, (New York, NY, USA), pp. 23–30, ACM, 2007.

[16] L. Azzopardi, M. de Rijke, and K. Balog, "Building simulated queries for known-item topics: An analysis using six european languages," in *Proceedings of the 30th International ACM SIGIR Conference.*, SIGIR '07, (New York, NY, USA), pp. 455–462, ACM, 2007.

[17] J. Kim and W. B. Croft, "Retrieval experiments using pseudo-desktop collections," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, (New York, NY, USA), pp. 1297–1306, ACM, 2009.

[18] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft, "Neural ranking models with weak supervision," in *Proceedings of The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[19] D. Vianna, A.-M. Yong, C. Xia, A. Marian, and T. Nguyen, "A tool for personal data extraction," in *Data Engineering Workshops (ICDEW) 2014 IEEE 30th International Conference on*, pp. 80–83, 2014.

[20] J. Gemmell, G. Bell, and R. Lueder, "Mylifebits: a personal database for everything," *Communications of the ACM*, vol. 49, no. 1, pp. 88–95, 2006.

[21] G. Bell and J. Gemmell, *Total Recall: How the E-Memory Revolution Will Change Everything*. Penguin, 2009.

[22] D. R. Karger, K. Bakshi, D. Huynh, D. Quan, and V. Sinha, "Haystack: A general-purpose information management tool for end users based on semistructured data," in *CIDR*, pp. 13–26, 2005.

[23] Z. Xu, M. Karlsson, C. Tang, and C. Karamanolis, "Towards a Semantic-Aware File Store," in *Proc. of the Workshop on Hot Topics in Operating Systems*, 2003.

[24] S. Zerr, E. Demidova, and S. Chernov, "deskweb2.0: Combining desktop and social search," in *Proc. of Desktop Search Workshop, In conjunction with the 33rd Annual International ACM SIGIR 2010*.

[25] X. Wang, M. Bendersky, D. Metzler, and M. Najork, "Learning to rank with selection bias in personal search," in *Proc. of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–124, 2016.

[26] H. Zamani, M. Bendersky, M. Zhang, and X. Wang, "Situational context for ranking in personal search," in *WWW*, 2017.

[27] M. Bendersky, X. Wang, D. Metzler, and M. Najork, "Learning from user interactions in personal search via attribute parameterization," in *Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 791–800, 2017.